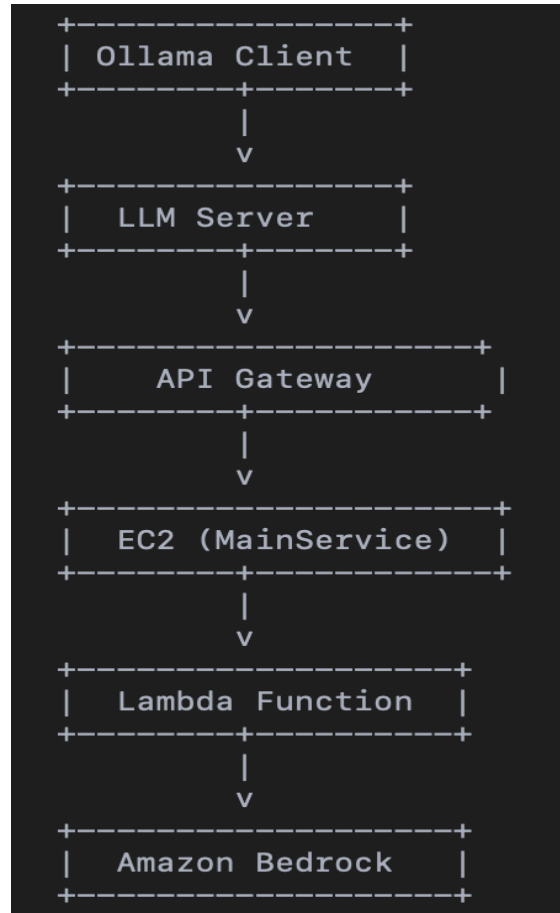


CS 441 Distributed Systems in Cloud Computing

Homework 3

Raunak Kumar Singh 675967946



Architecture Flow

1. Ollama Client:

- The Ollama Client initiates the process by sending a request (query) to the **LLM Server** for processing.
- This client can be any front-end application, user interface, or system that interacts with the backend, such as a web or mobile app.

2. LLM Server:

- The **LLM Server** handles incoming requests from the Ollama Client. It processes the request and prepares it for further handling by other components in the system.

- The **LLM Server** may serialize the data (for instance, creating a mock serialization of the request data as demonstrated in the `serializeRequest` method).
- The server may log structured data (like the query) for tracking and monitoring purposes.
- **Logging:** Logs the request being processed (i.e., Structured Data), prepares the response for the next component in the flow.

3. **API Gateway:**

- The **API Gateway** exposes HTTP endpoints and acts as a managed entry point for external requests from server.
- **Request Flow:** The request passes through the **API Gateway**, which routes the request to the **Lambda Function**.
- The API Gateway ensures secure and scalable handling of incoming API requests from external clients.

4. **MainService:**

- The **MainService** is an **EC2** service that acts as an intermediary layer between the **LLM Server** and AWS infrastructure. It performs necessary operations like request formatting, logging, validation, and managing workflows before invoking the **AWS Lambda Function**.
- It communicates with AWS services via HTTP, using AWS credentials, and logging responses.
- **Request Flow:** The processed request is passed to **MainService on EC2**, which invokes the **Lambda Function**.
- **Lambda Invocation:** MainService prepares a payload for invoking the Lambda function and handles the communication.

5. **Lambda Function:**

- The **Lambda Function** processes the incoming request further, which can include custom logic for data transformation, analysis, or forwarding.
- In your example, the **Lambda Function** interacts with **Amazon Bedrock**, passing the request and receiving responses.
- **Processing Flow:** The function processes the request, interacts with **Amazon Bedrock**, and returns a response back to the API Gateway.

6. Amazon Bedrock:

- **Amazon Bedrock** is the AI foundation service in AWS used for advanced language processing tasks. It processes the query using one of the available foundational models, such as those for text generation, summarization, or other tasks.
- **Response Flow:** Once Amazon Bedrock processes the request, it sends the response back through the **Lambda Function**, which forwards it back to the **EC2 MainService**

Key Components:

1. **Ollama Client:** Initiates the request with a query.
2. **LLM Server:** Receives the request and prepares it for processing, logs structured data.
3. **MainService:** Forwards the request, invokes Lambda, handles response.
4. **API Gateway:** Routes the request to Lambda and handles the external API.
5. **Lambda Function:** Processes the request, interacts with Amazon Bedrock.
6. **Amazon Bedrock:** Uses foundation models to process the query and generates the response.

The conversational file is uploaded on Github as part of the submission.